Finitary Winning in \omega-Regular Games

Krishnendu Chatterjee
Thomas A. Henzinger
Florian Horn

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2007-120
http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-120.html

October 8, 2007

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **08 OCT 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **Finitary Winning in omega-Regular Games**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Berkeley,Department of Electrical Engineering and Computer Sciences,Berkeley,CA,94720**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Games on graphs with !-regular objectives provide a model for the control and synthesis of reactive systems. Every !-regular objective can be decomposed into a safety part and a liveness part. The liveness part ensures that something good happens ?eventually.? Two main strengths of the classical, infinite-limit formulation of liveness are robustness (independence from the granularity of transitions) and simplicity (abstraction of complicated time bounds). However, the classical liveness formulation suffers from the drawback that the time until something good happens may be unbounded. A stronger formulation of liveness so-called finitary liveness, overcomes this drawback, while still retaining robustness and simplicity. Finitary liveness requires that there exists an unknown, fixed bound b such that something good happens within b transitions. While for one-shot liveness (reachability) objectives, classical and finitary liveness coincide, for repeated liveness (B?uchi) objectives the finitary formulation is strictly stronger. In this work we study games with finitary parity and Streett (fairness) objectives. We prove the determinacy of these games, present algorithms for solving these games, and characterize the memory requirements of winning strategies. We show that finitary parity games can be solved in polynomial time, which is not known for infinitary parity games. For finitary Streett games, we give an EXPTIME algorithm and show that the problem is NP-hard. Our algorithms can be used, for example, for synthesizing controllers that do not let the response time of a system increase without bound.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**29**	

Finitary Winning in ω-Regular Games[⋆,⋆⋆]

Krishnendu Chatterjee[1], Thomas A. Henzinger[1,2], and Florian Horn[3,4]

[1] University of California, Berkeley, USA
[2] EPFL, Switzerland
[3] RWTH, Aachen, Germany
[4] Liafa, Université Paris 7, France
{c_krish,tah}@eecs.berkeley.edu, horn@liafa.jussieu.fr

Abstract. Games on graphs with ω-regular objectives provide a model for the control and synthesis of reactive systems. Every ω-regular objective can be decomposed into a safety part and a liveness part. The liveness part ensures that something good happens "eventually." Two main strengths of the classical, infinite-limit formulation of liveness are robustness (independence from the granularity of transitions) and simplicity (abstraction of complicated time bounds). However, the classical liveness formulation suffers from the drawback that the time until something good happens may be unbounded. A stronger formulation of liveness, so-called *finitary* liveness, overcomes this drawback, while still retaining robustness and simplicity. Finitary liveness requires that there exists an unknown, fixed bound b such that something good happens within b transitions. While for one-shot liveness (reachability) objectives, classical and finitary liveness coincide, for repeated liveness (Büchi) objectives, the finitary formulation is strictly stronger. In this work we study games with finitary parity and Streett (fairness) objectives. We prove the determinacy of these games, present algorithms for solving these games, and characterize the memory requirements of winning strategies. We show that finitary parity games can be solved in polynomial time, which is not known for infinitary parity games. For finitary Streett games, we give an EXPTIME algorithm and show that the problem is NP-hard. Our algorithms can be used, for example, for synthesizing controllers that do not let the response time of a system increase without bound.

1 Introduction

Games played on graphs are suitable models for multi-component systems: vertices represent states; edges represent transitions; players represent components; and objectives represent specifications. The specification of a component is typically given as an ω-regular condition [15], and the resulting ω-regular games have been used for solving control and verification problems (see, e.g., [3, 18, 19]).

[⋆] This research was supported in part by the AFOSR MURI grant F49620-00-1-0327 and the NSF ITR grant CCR-0225610, and EU-TMR network Games.

[⋆⋆] The paper is a combined and extended version of the papers [5, 11].

Every ω-regular specification (indeed, every specification) can be decomposed into a safety part and a liveness part [1]. The safety part ensures that the component will not do anything "bad" (such as violate an invariant) within any finite number of transitions. The liveness part ensures that the component will do something "good" (such as proceed, or respond, or terminate) within some finite number of transitions. Liveness can be violated only in the limit, by infinite sequences of transitions, as no bound is stipulated on when the "good" thing must happen. This infinitary, classical formulation of liveness has both strengths and weaknesses. A main strength is robustness, in particular, independence from the chosen granularity of transitions. Another main strength is simplicity, allowing liveness to serve as an abstraction for complicated safety conditions. For example, a component may always respond in a number of transitions that depends, in some complicated manner, on the exact size of the stimulus. Yet for correctness, we may be interested only that the component will respond "eventually." However, these strengths also point to a weakness of the classical definition of liveness: it can be satisfied by components that in practice are quite unsatisfactory because no bound can be put on their response time. It is for this reason that alternative, stronger formulations of liveness have been proposed. One of these is *finitary* liveness [2, 6]: finitary liveness does not insist on response within a known bound b (i.e., every stimulus is followed by a response within b transitions), but on response within some unknown bound (i.e., there exists b such that every stimulus is followed by a response within b transitions). Note that in the finitary case, the bound b may be arbitrarily large, but the response time must not grow forever from one stimulus to the next. In this way, finitary liveness still maintains the robustness (independence of step granularity) and simplicity (abstraction of complicated safety) of traditional liveness, while removing unsatisfactory implementations.

In this paper, we study graph games with finitary winning conditions. The motivation is the same as for finitary liveness. Consider, for example, the synthesis of an elevator controller as a strategy in a game where one player represents the environment (i.e., the pushing of call buttons on various floors, and the pushing of target buttons inside the elevators), and the other player represents the elevator control (i.e., the commands to move an elevator up or down, and the opening and closing of elevator doors). Clearly, one objective of the controller is that whenever a call button is pushed on a floor, then an elevator will eventually arrive, and whenever a target button is pushed inside an elevator, then the elevator will eventually get to the corresponding floor. Note that this objective is formulated in an infinitary way (the key term is "eventually"). This is because, for robustness and simplicity, we do not wish to specify for each state the exact number of transitions until the objective must be met. However, a truly unbounded implementation of elevator control (where the response time grows from request to request, without bound) would be utterly unsatisfactory. A finitary interpretation of the objective prohibits such undesirable control strategies: there must exist a bound b such that the controller meets every call request, and every target request, within b transitions.

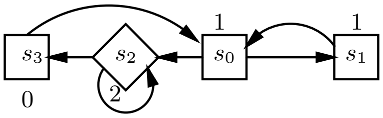

Fig. 1. A simple game graph.

We formalize finitary winning for the normal form of ω-regular objectives called *parity* conditions [20]. A parity objective assigns a non-negative integer priority to every vertex, and the objective of player 1 is to make sure that the lowest priority that repeats infinitely often is even. This is an infinitary objective, as player 1 can win by ensuring that every odd priority that repeats infinitely often is followed by a smaller even priority "eventually" (arbitrarily many transitions later). The *finitary* parity objective, by contrast, insists that player 1 ensures that there exists a bound b such that every odd priority that repeats infinitely often is followed by a smaller even priority within b transitions. The finitary parity objective is stronger than the classical parity objective, as is illustrated by the following example.

Example 1. Consider the game shown in Figure 1. The square-shaped states are player 1 states, where player 1 chooses the successor state, and the diamond-shaped states are player 2 states (we will follow this convention throughout this paper). The priorities of states are shown next to each state in the figure. If player 1 follows a memoryless strategy σ that chooses the successor s_2 at state s_0, this ensures that against all strategies π for player 2, the minimum priority of the states that are visited infinitely often is even (either state s_3 is visited infinitely often, or both states s_0 and s_1 are visited finitely often). However, consider the strategy π_w for player 2: the strategy π_w is played in rounds, and in round $k \geq 0$, whenever player 1 chooses the successor s_2 at state s_0, player 2 stays in state s_2 for k transitions, and then goes to state s_3 and proceeds to round $k+1$. The strategy π_w ensures that for all strategies σ for player 1, either the minimum priority visited infinitely often is 1 (i.e., both states s_0 and s_1 are visited infinitely often and state s_3 is visited finitely often); or states of priority 1 are visited infinitely often, and the distances between visits to states of priority 1 and subsequent visits to states of priority 0 increase without bound (i.e., the limit of the distances is ∞). Hence it follows that in this game, although player 1 can win for the parity objective, she cannot win for the finitary parity objective. ∎

We prove that games with finitary parity objectives are determined: for every state either there is a player 1 strategy (a winning strategy for player 1) that ensures that the finitary parity objective is satisfied against all player 2 strategies, or there is a player 2 strategy (a winning strategy for player 2) that ensures that the finitary parity objective is violated against all player 1 strategies. Similar to games with infinitary parity objectives, we establish the existence of winning strategies that are *memoryless* (independent of the history of the play) for player 1. However, winning strategies for player 2 in general require infinite memory; this is in contrast to infinitary parity objectives, where memoryless winning strategies exist also for player 2 [8]. Thus the analysis of finitary parity objectives is more involved. We present a polynomial-time algorithm that computes the winning states of finitary parity games in time $O(n^2 \cdot m)$ for game graphs with n states and m edges. Again this is in contrast to classical, infinitary parity games, for which no polynomial-time algorithm is known (the best known algorithms have time complexity $O(n^{\lfloor \frac{d}{2} \rfloor} \cdot m)$ [12] or $n^{O(\sqrt{n})}$ [13]).

In addition to finitary parity, we study finitary Streett objectives. Streett objectives require that if some stimuli are repeated infinitely often, then the corresponding responses occur infinitely often. The finitary interpretation requires, in addition, that there exists a bound b on all required response times (i.e., on the number of transitions between stimulus and corresponding response). We present an algorithm for games with finitary Streett objectives that computes the winning sets in time $O(n^2 \cdot m \cdot d \cdot 2^d)$ for game graphs with n states, m edges, and finitary Streett objectives with d pairs. Hence, the winning states can be decided in EXPTIME. We also show that deciding if a given state is winning for player 1 is NP-hard. For comparison, the decision problem for games with infinitary Streett objectives is coNP-complete [8], and the winning states can be computed in time $O(n^d \cdot d! \cdot m)$ [10]. For classical as well as finitary Streett games, finite-memory winning strategies exist for player 1: for infinitary Streett objectives, winning strategies require $d!$ memory [7, 10]; for finitary Streett objectives, we show an upper bound of $d \cdot 2^d$ and a lower bound of $2^{\lfloor \frac{d}{2} \rfloor}$ for the memory requirement, for (finitary) Streett objectives with d pairs. However, while in the classical case memoryless winning strategies exist for player 2 [8], in the finitary case the winning strategies for player 2 may require infinite memory.

We focus on finitary parity and Streett objectives. The finitary parity objectives are a canonical form to express finitary versions of ω-regular objectives; they subsume finitary reachability, finitary Büchi, and finitary co-Büchi objectives as special cases. The Streett objectives capture liveness conditions that are of particular interest in system design, as they correspond to strong fairness (compassion) constraints [15]. The finitary Streett objectives, therefore, give the finitary formulation of strong fairness.

2 Games with ω-Regular Objectives

2.1 Game graphs

Game graphs. A *game graph* $G = ((S, E), (S_1, S_2))$ consists of a directed graph (S, E) with a finite state space S and a set E of edges, and a partition (S_1, S_2) of the state space S into two sets. The states in S_1 are player 1 states, and the states in S_2 are player 2 states. For a state $s \in S$, we write $E(s) = \{t \in S \mid (s, t) \in E\}$ for the set of successor states of s. We assume that every state has at least one out-going edge, i.e., $E(s)$ is non-empty for all states $s \in S$.

Plays. A game is played by two players: player 1 and player 2, who form an infinite path in the game graph by moving a token along edges. They start by placing the token on an initial state, and then they take moves indefinitely in the following way. If the token is on a state in S_1, then player 1 moves the token along one of the edges going out of the state. If the token is on a state in S_2, then player 2 does likewise. The result is an infinite path in the game graph; we refer to such infinite paths as plays. Formally, a *play* is an infinite sequence $\langle s_0, s_1, s_2, \ldots \rangle$ of states such that $(s_k, s_{k+1}) \in E$ for all $k \geq 0$. We write Ω for the set of all plays.

Strategies. A strategy for a player is a recipe that specifies how to extend plays. Formally, a *strategy* σ for player 1 is a function $\sigma\colon S^* \cdot S_1 \to S$ that, given a finite sequence of states (representing the history of the play so far) which ends in a player 1 state, chooses the next state. The strategy must choose only available successors, i.e., for all $w \in S^*$ and $s \in S_1$, if $\sigma(w \cdot s) = t$, then $t \in E(s)$. The strategies for player 2 are defined analogously. We write Σ and Π for the sets of all strategies for player 1 and player 2, respectively.

An equivalent definition of strategies is as follows. Let M be a set called *memory.* A strategy with memory can be described as a pair of functions: (a) a *memory-update* function $\sigma_u\colon S \times M \to M$ that, given the memory and the current state, updates the memory; and (b) a *next-state* function $\sigma_n\colon S \times M \to S$ that, given the memory and the current state, specifies the successor state. The strategy is *finite-memory* if the memory M is finite and for a finite-memory strategy σ we write $|\sigma|$ to denote the size of its memory, i.e., $|M|$. The strategy is *memoryless* if the memory M is a singleton set. The memoryless strategies do not depend on the history of a play, but only on the current state. Each memoryless strategy for player 1 can be specified as a function $\sigma\colon S_1 \to S$ such that $\sigma(s) \in E(s)$ for all $s \in S_1$, and analogously for memoryless player 2 strategies. Given a starting state $s \in S$, a strategy $\sigma \in \Sigma$ for player 1, and a strategy $\pi \in \Pi$ for player 2, there is a unique play, denoted $\omega(s, \sigma, \pi) = \langle s_0, s_1, s_2, \ldots \rangle$, which is defined as follows: $s_0 = s$ and for all $k \geq 0$, if $s_k \in S_1$, then $\sigma(s_0, s_1, \ldots, s_k) = s_{k+1}$, and if $s_k \in S_2$, then $\pi(s_0, s_1, \ldots, s_k) = s_{k+1}$.

Counting strategies. We call an infinite memory strategy σ *finite-memory counting* if there is a finite-memory strategy σ' such that for all $j \geq 0$ there exists $k \leq j$ such that the following condition hold: for all $w \in S^*$ such that $|w| = j$ and for all $s \in S_1$ we have $\sigma(w \cdot s) = \sigma'(\mathrm{suffix}(w, k) \cdot s)$, where for $w \in S^*$ of length j and $k \leq j$ we denote by $\mathrm{suffix}(w, k)$ the suffix of w of length k. In other words, the strategy σ repeatedly plays the finite-memory strategy σ' in different segments of the play and the switch of the strategy in different segments only depends on the length of the play. We denote by $\mathrm{count}(|\sigma|)$ the size of the memory of the finite-memory strategy σ', i.e., $\mathrm{count}(|\sigma|) = |\sigma'|$. We use similar notations for player 2 strategies.

2.2 Classical winning conditions

We first define the class of ω-regular objectives and the classical notion of winning.

Objectives. Objectives for the players in non-terminating games are specified by providing the sets $\Phi, \Psi \subseteq \Omega$ of *winning plays* for player 1 and player 2, respectively. We consider zero-sum games, where the objectives of both players are complementary, i.e., $\Psi = \Omega \setminus \Phi$. The class of *$\omega$-regular objectives* [20] are of special interest since they form a robust class of objectives for verification and synthesis. The ω-regular objectives, and subclasses thereof, can be specified in the following forms. For a play $\omega = \langle s_0, s_1, s_2, \ldots \rangle \in \Omega$, we define $\mathrm{Inf}(\omega) = \{s \in S \mid s_k = s \text{ for infinitely many } k \geq 0\}$ to be the set of states that occur infinitely often in ω.

1. *Reachability and safety objectives.* Given a set $F \subseteq S$ of states, the reachability objective $\mathrm{Reach}(F)$ requires that some state in F be visited, and dually, the safety objective $\mathrm{Safe}(F)$ requires that only states in F be visited. Formally, the sets of winning plays are $\mathrm{Reach}(F) = \{\langle s_0, s_1, s_2, \ldots \rangle \in \Omega \mid \exists k \geq 0.\ s_k \in F\}$ and $\mathrm{Safe}(F) = \{\langle s_0, s_1, s_2, \ldots \rangle \in \Omega \mid \forall k \geq 0.\ s_k \in F\}$.
2. *Büchi and co-Büchi objectives.* Given a set $F \subseteq S$ of states, the Büchi objective $\mathrm{Buchi}(F)$ requires that some state in F be visited infinitely often, and dually, the co-Büchi objective $\mathrm{coBuchi}(F)$ requires that only states in F be visited infinitely often. Thus, the sets of winning plays are $\mathrm{Buchi}(F) = \{\omega \in \Omega \mid \mathrm{Inf}(\omega) \cap F \neq \emptyset\}$ and $\mathrm{coBuchi}(F) = \{\omega \in \Omega \mid \mathrm{Inf}(\omega) \subseteq F\}$.
3. *Rabin and Streett objectives.* Given a set $P = \{(E_1, F_1), \ldots, (E_d, F_d)\}$ of pairs of sets of states (i.e, for all $1 \leq j \leq d$, both $E_j \subseteq S$ and $F_j \subseteq S$), the Rabin objective $\mathrm{Rabin}(P)$ requires that for some pair $1 \leq j \leq d$, all states in E_j be visited finitely often, and some state in F_j be visited infinitely often. Hence, the winning plays are $\mathrm{Rabin}(P) = \{\omega \in \Omega \mid \exists 1 \leq j \leq d.\ (\mathrm{Inf}(\omega) \cap E_j = \emptyset \text{ and } \mathrm{Inf}(\omega) \cap F_j \neq \emptyset)\}$. Dually, given $P = \{(E_1, F_1), \ldots, (E_d, F_d)\}$, the Streett objective $\mathrm{Streett}(P)$ requires that for all pairs $1 \leq j \leq d$, if some state in F_j is visited infinitely often, then some state in E_j be visited infinitely often, i.e., $\mathrm{Streett}(P) = \{\omega \in \Omega \mid \forall 1 \leq j \leq d.\ (\mathrm{Inf}(\omega) \cap E_j \neq \emptyset \text{ or } \mathrm{Inf}(\omega) \cap F_j = \emptyset)\}$.
4. *Parity objectives.* Given a function $p\colon S \to \{0, 1, 2, \ldots, d-1\}$ that maps every state to an integer *priority*, the parity objective $\mathrm{Parity}(p)$ requires that of the states that are visited infinitely often, the least priority be even. Formally, the set of winning plays is $\mathrm{Parity}(p) = \{\omega \in \Omega \mid \min\{p(\mathrm{Inf}(\omega))\} \text{ is even}\}$. The dual, co-parity objective has the set $\mathrm{coParity}(p) = \{\omega \in \Omega \mid \min\{p(\mathrm{Inf}(\omega))\} \text{ is odd}\}$ of winning plays. Parity objectives are closed under complementation: given a function $p : S \to \{0, 1, \ldots, d-1\}$, consider the function $p+1 : S \to \{1, 2, \ldots, d\}$ defined as $p+1(s) = p(s)+1$, for all $s \in S$, and then we have $\mathrm{Parity}(p+1) = \mathrm{coParity}(p)$.

Every parity objective is both a Rabin objective and a Streett objective. Hence, the parity objectives are closed under complementation. The Büchi and co-Büchi objectives are special cases of parity objectives with two priorities, namely, $p\colon S \to \{0, 1\}$ for Büchi objectives with $F = p^{-1}(0)$, and $p\colon S \to \{1, 2\}$ for co-Büchi objectives with $F = p^{-1}(2)$. The reachability and safety objectives can be turned into Büchi and co-Büchi objectives, respectively, on slightly modified game graphs.

Winning. Given an objective $\Phi \subseteq \Omega$ for player 1, a strategy $\sigma \in \Sigma$ is a *winning strategy* for player 1 from a set $U \subseteq S$ of states if for all player 2 strategies $\pi \in \Pi$ and all states $s \in U$, the play $\omega(s, \sigma, \pi)$ is winning, i.e., $\omega(s, \sigma, \pi) \in \Phi$. The winning strategies for player 2 are defined analogously. A state $s \in S$ is winning for player 1 with respect to the objective Φ if player 1 has a winning strategy from $\{s\}$. Formally, the set of *winning states* for player 1 with respect to the objective Φ is $W_1(\Phi) = \{s \in S \mid \exists \sigma \in \Sigma.\ \forall \pi \in \Pi.\ \omega(s, \sigma, \pi) \in \Phi\}$. Analogously, the set of winning states for player 2 with respect to an objective $\Psi \subseteq \Omega$ is $W_2(\Psi) = \{s \in S \mid \exists \pi \in \Pi.\ \forall \sigma \in \Sigma.\ \omega(s, \sigma, \pi) \in \Psi\}$. We say that there exists a (memoryless; finite-memory) winning strategy for player 1 with respect to the objective Φ if there exists such a strategy from the set $W_1(\Phi)$; and similarly for player 2.

Theorem 1 (Classical determinacy and strategy complexity).

1. [9] *For all game graphs, all Rabin objectives Φ for player 1, and the complementary Streett objective $\Psi = \Omega \backslash \Phi$ for player 2, we have $W_1(\Phi) = S \backslash W_2(\Psi)$.*
2. [8] *For all game graphs and all Rabin objectives for player 1, there exists a memoryless winning strategy for player 1.*
3. [9] *For all game graphs and all Streett objectives for player 2, there exists a finite-memory winning strategy for player 2. However, in general no memoryless winning strategy exists.*

3 Finitary Winning Conditions

We now define a stronger notion of winning, namely, *finitary winning*, in games with parity and Streett objectives.

3.1 Finitary winning for parity objectives

For parity objectives, the finitary winning notion requires that for each visit to an odd priority that is visited infinitely often, the distance to a stronger (i.e., lower) even priority be bounded. To define the winning plays formally, we need the concept of a distance sequence.

Distance sequences for parity objectives. Given a play $\omega = \langle s_0, s_1, s_2, \ldots \rangle$ and a priority function $p\colon S \to \{0, 1, \ldots, d-1\}$, we define a sequence of distances $dist_k(\omega, p)$, for all $k \geq 0$, as follows:

$$dist_k(\omega, p) = \begin{cases} 0 & \text{if } p(s_k) \text{ is even;} \\ \inf\{k' \geq k \mid p(s_{k'}) \text{ is even and } p(s_{k'}) < p(s_k)\} & \text{if } p(s_k) \text{ is odd.} \end{cases}$$

Intuitively, the distance for a position k in a play with an odd priority at position k, denotes the shortest distance to a stronger even priority in the play. We assume the standard convention that the infimum of the empty set is ∞.

Finitary parity objectives. The finitary parity objective $\mathrm{finParity}(p)$ for a priority function p requires that the sequence of distances for the positions with odd priorities that occur infinitely often be bounded. This is equivalent to requiring that the sequence of all distances be bounded in the limit, and captures the notion that the "good" (even) priorities that appear infinitely often do not appear infinitely rarely. Formally, the sets of winning plays for the finitary parity objective and its complement are $\mathrm{finParity}(p) = \{\omega \in \Omega \mid \limsup_{k\to\infty} \mathit{dist}_k(\omega, p) < \infty\}$ and $\mathrm{infParity}(p) = \{\omega \in \Omega \mid \limsup_{k\to\infty} \mathit{dist}_k(\omega, p) = \infty\}$, respectively. Observe that if a play ω is winning for a co-parity objective, then the lim sup of the distance sequence for ω is ∞, that is, $\mathrm{coParity}(p) \subseteq \mathrm{infParity}(p)$. However, if a play ω is winning for a (classical) parity objective, then the lim sup of the distance sequence for ω can be ∞ (as shown in Example 1), that is, $\mathrm{finParity}(p) \subsetneq \mathrm{Parity}(p)$. Given a game graph G and a priority function p, solving the finitary parity game requires computing the two winning sets $W_1(\mathrm{finParity}(p))$ and $W_2(\mathrm{infParity}(p))$.

Remark 1. Recall that Büchi and co-Büchi objectives correspond to parity objectives with two priorities. A finitary Büchi objective is in general a strict subset of the corresponding classical Büchi objective; a finitary co-Büchi objective coincides with the corresponding classical co-Büchi objective. However, it can be shown that for parity objectives with two priorities, the classical winning sets and the finitary winning sets are the same; that is, for all game graphs G and all priority functions p with two priorities, we have $W_1(\mathrm{finParity}(p)) = W_1(\mathrm{Parity}(p))$ and $W_2(\mathrm{infParity}(p)) = W_2(\mathrm{coParity}(p))$. Note that in Example 1, we have $s_0 \in W_1(\mathrm{Parity}(p))$ and $s_0 \not\in W_1(\mathrm{finParity}(p))$. This shows that for priority functions with three or more priorities, the winning set for a finitary parity objective can be a strict subset of the winning set for the corresponding classical parity objective, that is, $W_1(\mathrm{finParity}(p)) \subsetneq W_1(\mathrm{Parity}(p))$. ∎

3.2 Weak parity and bounded parity objectives

We now define *weak parity* objectives and the *bounded parity* objectives. We will later use the solution of weak parity objectives iteratively to solve games with bounded parity objectives, and then use the solution of bounded parity objectives iteratively to solve games with finitary parity objectives.

Weak parity objectives. In a weak parity objective the winner of a play is decided by considering the minimum priority state that appear in the play: if the minimum priority is even, then player 1 wins, and otherwise player 2 is the winner. For a play $\omega = \langle s_0, s_1, s_2, \ldots \rangle \in \Omega$, we define $\mathrm{Occur}(\omega) = \{s \in S \mid s_k = s \text{ for some } k \geq 0\}$ to be the set of states that occur in ω. For a priority function p, the weak parity objective $\mathrm{weakParity}(p)$ and its complement $\mathrm{weakcoParity}(p)$ are defined as follows:

$$\begin{aligned} \mathrm{weakParity}(p) &= \{\omega \in \Omega \mid \min(p(\mathrm{Occur}(\omega))) \text{ is even}\}; \\ \mathrm{weakcoParity}(p) &= \{\omega \in \Omega \mid \min(p(\mathrm{Occur}(\omega))) \text{ is odd}\}. \end{aligned}$$

Bounded parity objectives. The bounded parity objective requires the distance sequence to be bounded, and its complement requires the distance sequence to be unbounded. For a priority function p, the bounded parity objective $\mathrm{bndParity}(p)$ and its complement $\mathrm{unbndParity}(p)$ are defined as follows:

$$\begin{aligned}\mathrm{bndParity}(p) &= \{\omega \in \Omega \mid \exists j \in \mathbb{N}.\ \forall k \geq 0.\ \mathit{dist}_k(\omega, p) \leq j\} \\ &= \{\omega \in \Omega \mid \sup\{\mathit{dist}_k(\omega, p) \mid k \geq 0\} < \infty\}; \\ \mathrm{unbndParity}(p) &= \{\omega \in \Omega \mid \forall j \in \mathbb{N}.\ \exists k \geq 0.\ \mathit{dist}_k(\omega, p) \geq j\} \\ &= \{\omega \in \Omega \mid \sup\{\mathit{dist}_k(\omega, p) \mid k \geq 0\} = \infty\}.\end{aligned}$$

Relationship between objectives. We already noted in Remark 1 that in general we have

$$\mathrm{finParity}(p) \subsetneq \mathrm{Parity}(p); \qquad W_1(\mathrm{finParity}(p)) \subsetneq W_1(\mathrm{Parity}(p)).$$

Consider a play ω such that $\omega \in \mathrm{bndParity}(p)$, then there exists a $j \in \mathbb{N}$ such that for all $k \geq 0$ we have $\mathit{dist}_k(\omega, p) \leq j$, and hence $\limsup_{k\to\infty} \mathit{dist}_k(\omega, p) \leq j$. Hence we have $\mathrm{bndParity}(p) \subseteq \mathrm{finParity}(p)$. However, consider a play ω such that the priority sequence of ω is $1 \cdot 2^\omega$, then we have $\omega \in \mathrm{finParity}(p)$, but $\mathit{dist}_0(\omega, p) = \infty$, and thus $\omega \notin \mathrm{bndParity}(p)$. Moreover, a graph with a starting state s_0 with priority 1 and an edge to a state s_1 such that s_1 has a self-loop and priority 2, shows that in general we have $W_1(\mathrm{bndParity}(p)) \subsetneq W_1(\mathrm{finParity}(p))$. Thus we obtain the following relationship:

$$\mathrm{bndParity}(p) \subsetneq \mathrm{finParity}(p) \subsetneq \mathrm{Parity}(p);$$

$$W_1(\mathrm{bndParity}(p)) \subsetneq W_1(\mathrm{finParity}(p)) \subsetneq W_1(\mathrm{Parity}(p)).$$

Also observe that for a play $\omega \in \mathrm{bndParity}(p)$ the minimum priority that appears in ω must be even, otherwise if the minimum priority is odd, then the position where the minimum odd priority occurs will have distance ∞. Thus we have $\mathrm{bndParity}(p) \subseteq \mathrm{weakParity}(p)$. Consider a play ω such that the priority sequence of ω is $0 \cdot 1^\omega$, then $\omega \in \mathrm{weakParity}(p)$, however, $\omega \notin \mathrm{bndParity}(p)$. Hence we have the following relationship:

$$\mathrm{bndParity}(p) \subsetneq \mathrm{weakParity}(p); \qquad W_1(\mathrm{bndParity}(p)) \subsetneq W_1(\mathrm{weakParity}(p)).$$

The objective $\mathrm{weakParity}(p)$ is incomparable in terms of inclusion to $\mathrm{finParity}(p)$ and $\mathrm{Parity}(p)$. Consider a play ω with the priority sequence $1 \cdot 2^\omega$, then $\omega \in \mathrm{finParity}(p)$ and $\omega \in \mathrm{Parity}(p)$, however, $\omega \notin \mathrm{weakParity}(p)$. On the other hand, consider a play ω with the priority sequence $0 \cdot 1^\omega$, then $\omega \in \mathrm{weakParity}(p)$, however, $\omega \notin \mathrm{Parity}(p)$ and $\omega \notin \mathrm{finParity}(p)$.

3.3 Finitary winning for Streett objectives

The notion of distance sequence for parity objectives has a natural extension to Streett objectives.

Distance sequences for Streett objectives. Given a play $\omega = \langle s_0, s_1, s_2, \ldots \rangle$ and a set $P = \{(E_1, F_1), \ldots, (E_d, F_d)\}$ of Streett pairs of state sets, the d sequences of distances $dist^j_k(\omega, P)$, for all $k \geq 0$ and $1 \leq j \leq d$, are defined as follows:

$$dist^j_k(\omega, P) = \begin{cases} 0 & \text{if } s_k \notin F_j; \\ \inf\{k' \geq k \mid s_{k'} \in E_j\} & \text{if } s_k \in F_j. \end{cases}$$

Let $dist_k(\omega, P) = \max\{dist^j_k(\omega, P) \mid 1 \leq j \leq d\}$ for all $k \geq 0$.

Finitary Streett objectives. The finitary Streett objective $\text{finStreett}(P)$ for a set P of Streett pairs requires that the distance sequence be bounded in the limit, i.e., the winning plays are $\text{finStreett}(P) = \{\omega \in \Omega \mid \limsup_{k\to\infty} dist_k(\omega, P) < \infty\}$. We use the following notations for the complementary objective: $\text{infStreett}(P) = \Omega \setminus \text{finStreett}(P)$.

Example 2. Consider the game graph of Figure 2. Player 2 generates requests of type Req_1 and Req_2; these are shown as labeled edges in the figure. Player 1 services a request of type Req_i by choosing an edge labeled $Serv_i$, for $i = 1, 2$. Whenever a request is received, further requests of the same type are disabled until the request is serviced; then the requests of this type are enabled again. The state s_0 represents the case when there are no unserviced requests; the states s_1 and s_2 represent the cases when there are unserviced requests of type Req_1 and Req_2, respectively; and the states s_7 and s_8 represent the cases when there are unserviced requests of both types, having arrived in either order. On arrival of a request of type Req_i, a state in F_i is visited, and when a request of type Req_i is serviced, a state in E_i is visited, for $i = 1, 2$. Hence $F_1 = \{s_1, s_8\}$, $F_2 = \{s_2, s_7\}$, $E_1 = \{s_5, s_{12}\}$, and $E_2 = \{s_6, s_{11}\}$. The Streett objective $\text{Streett}(P)$ with $P = \{(E_1, F_1), (E_2, F_2)\}$ requires that if a request of type Req_i is received infinitely often, then it be serviced infinitely often, for both $i = 1, 2$. The player 1 strategy $s_9 \to s_{11}$ and $s_{10} \to s_{12}$ is a *stack* strategy, which always services first the request type received last. The player 1 strategy $s_9 \to s_{12}$ and $s_{10} \to s_{11}$ is a *queue* strategy, which always services first the request type received first. Both the stack strategy and the queue strategy ensure that the classical Streett objective $\text{Streett}(P)$ is satisfied. However, for the stack strategy, the number of transitions between the arrival of a request of type Req_i and its service can be unbounded. Hence the stack strategy is not a winning strategy for player 1 with respect to the finitary Streett objective $\text{finStreett}(P)$. The queue strategy, by contrast, ensures not only that every request that is received infinitely often is serviced, but it also ensures that the number of transitions between the arrival of a request and its service is at most 6. Thus the queue strategy is winning for player 1 with respect to $\text{finStreett}(P)$. ∎

3.4 Weak Streett and bounded Streett objectives

We now define weak Streett objectives and bounded Streett objectives. We will later use the solution of games with bounded Streett objectives to solve games with finitary Streett objectives.

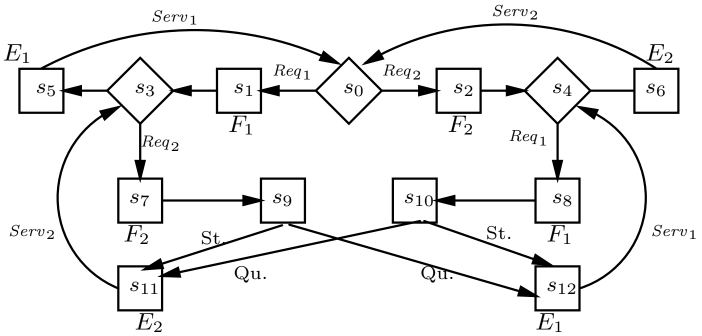

Fig. 2. A request-service game graph.

Weak Streett objectives. Similar to weak parity objectives, in weak Streett objectives the winner is decided considering the set of states that appear in a play. Given $P = \{(E_1, F_1), \dots, (E_d, F_d)\}$, the weak Streett objective weakStreett(P) requires that for all pairs $1 \le j \le d$, if some state in F_j is visited, then some state in E_j be visited, i.e.,

$$\mathrm{weakStreett}(P) = \{\omega \in \Omega \mid \forall 1 \le j \le d.\ (\mathrm{Occur}(\omega) \cap E_j \neq \emptyset \text{ or } \mathrm{Occur}(\omega) \cap F_j = \emptyset)\}.$$

Bounded Streett objectives. Similar to bounded parity objectives the bounded Streett objectives requires the distance sequence to be bounded. Formally, given $P = \{(E_1, F_1), \dots, (E_d, F_d)\}$, the bounded Streett objective is defined as follows:

$$\begin{aligned} \mathrm{bndStreett}(P) &= \{\omega \in \Omega \mid \exists j \in \mathbb{N}.\ \forall k \ge 0.\ dist_k(\omega, P) \le j\} \\ &= \{\omega \in \Omega \mid \sup\{dist_k(\omega, P) \mid k \ge 0\} < \infty\}. \end{aligned}$$

We use the following notations for the complementary objective: $\mathrm{unbndStreett}(P) = \Omega \setminus \mathrm{bndStreett}(P)$.

4 Finitary Parity Games: Determinacy and Complexity

We present an algorithm to solve games with finitary parity objectives. The correctness argument for the algorithm also proves determinacy for finitary parity games.[1] The algorithm is obtained by iteratively solving games with bounded parity objectives, and the solution of bounded parity objectives is obtained by iteratively solving games with weak parity objectives. We start with some preliminary notation and facts that will be required for the analysis of the algorithm.

[1] The determinacy of games with finitary parity objectives can also be proved by reduction to Borel objectives, using the determinacy of Borel games [16]; however, our proof is direct.

Closed sets. A set $U \subseteq S$ of states is a *closed set* for player 2 if the following two conditions hold: (a) for all states $u \in (U \cap S_2)$, we have $E(u) \subseteq U$, i.e., all successors of player 2 states in U are again in U; and (b) for all $u \in (U \cap S_1)$, we have $E(u) \cap U \neq \emptyset$, i.e., every player 1 state in U has a successor in U. The closed sets for player 1 are defined analogously. Every closed set U for player ℓ, for $\ell \in \{1, 2\}$, induces a sub-game graph, denoted $G \upharpoonright U$. For winning sets W_1 and W_2, we write W_1^G and W_2^G to explicitly specify the game graph G.

Proposition 1. *Consider a game graph G, and a closed set U for player 2. For every objective Φ for player 1, we have $W_1^{G \upharpoonright U}(\Phi) \subseteq W_1^G(\Phi)$.*

Attractors. Given a game graph G, a set $U \subseteq S$ of states, and a player $\ell \in \{1, 2\}$, the set $Attr_\ell(U, G)$ contains the states from which player ℓ has a strategy to reach a state in U against all strategies of the other player; that is, $Attr_\ell(U, G) = W_\ell^G(\text{Reach}(U))$. The set $Attr_1(U, G)$ can be computed inductively as follows: let $R_0 = U$; let $R_{i+1} = R_i \cup \{s \in S_1 \mid E(s) \cap R_i \neq \emptyset\} \cup \{s \in S_2 \mid E(s) \subseteq R_i\}$ for all $i \geq 0$; then $Attr_1(U, G) = \bigcup_{i \geq 0} R_i$. The inductive computation of $Attr_2(U, G)$ is analogous. For all states $s \in Attr_1(U, G)$, define $rank(s) = i$ if $s \in R_i \setminus R_{i-1}$, that is, $rank(s)$ denotes the least $i \geq 0$ such that s is included in R_i. Define a memoryless strategy $\sigma \in \Sigma$ for player 1 as follows: for each state $s \in (Attr_1(U, G) \cap S_1)$ with $rank(s) = i$, choose a successor $\sigma(s) \in (R_{i-1} \cap E(s))$ (such a successor exists by the inductive definition). It follows that for all states $s \in Attr_1(U, G)$ and all strategies $\pi \in \Pi$ for player 2, the play $\omega(s, \sigma, \pi)$ reaches U in at most $|Attr_1(U, G)|$ transitions.

Proposition 2. *For all game graphs G, all players $\ell \in \{1, 2\}$, and all sets $U \subseteq S$ of states, the set $S \setminus Attr_\ell(U, G)$ is a closed set for player ℓ.*

4.1 Solving games with weak parity objectives

We first informally describe an algorithm to solve games with weak parity objectives; the formal description of the complete algorithm is available in [14] and a detailed running time analysis is available in [4]. The algorithm takes as input a game graph G and a priority function p, and proceeds as follows: first it computes the player 1 attractor to the set $p^{-1}(0)$ of states with priority 0, and identifies the set $W_0 = Attr_1(p^{-1}(0), G)$ as a subset of $W_1^G(\text{weakParity}(p))$. Clearly in W_0 player 1 can play a memoryless attractor strategy to reach $p^{-1}(0)$ and ensure to win. Then $S \setminus W_0$ is a closed set for player 1 and induces a sub-game $G_1 = G \upharpoonright (S \setminus W_0)$. Then player 2 attractor is computed to the set $p^{-1}(1)$ in G_1 (i.e., attractor to the set of states with priority 1 in G_1) and the set $W_1 = Attr_2(p^{-1}(1) \cap (S \setminus W_0), G_1)$ is identified as a subset of $W_2^G(\text{weakcoParity}(p))$. Since G_1 is a closed set for player 1, a memoryless attractor strategy for player 2 in W_1 to reach $p^{1-}(1) \cap (S \setminus W_0)$ and stay safe in G_1 ensures that player 2 can satisfy weakcoParity(p) in W_1. The algorithm then removes the set W_1 from G_1 and proceeds on the sub-game. Finally the algorithm correctly obtains the set $W_1^G(\text{weakParity}(p))$ and $W_2^G(\text{weakcoParity}(p))$. In the set $W_1^G(\text{weakParity}(p))$ every odd priority state belongs to the attractor of a smaller even priority, and in the set $W_2^G(\text{weakcoParity}(p))$ every even priority state belongs to the attractor of a smaller odd priority. The winning strategies of the players in their respective winning sets can be obtained by composing memoryless attractor strategies. We now summarize the results on games with weak parity objectives.

Theorem 2 (Weak parity games[14, 4]). *For all game graphs $G = ((S, E), (S_1, S_2))$ and all priority functions p the following assertions hold.*

1. (Determinacy). *We have $W_1(\text{weakParity}(p)) = S \setminus W_2(\text{weakcoParity}(p))$.*
2. (Strategy complexity). *There exists a memoryless winning strategy for player 1 for objective* weakParity(p) *and there exists a memoryless winning strategy for player 2 for objective* weakcoParity(p)*.*
3. (Time complexity). *The sets $W_1(\text{weakParity}(p))$ and $W_2(\text{weakcoParity}(p))$ can be computed in $O(m)$ time, where $m = |E|$.*

4.2 Solving games with bounded parity objectives

In this section we will show how the solution of games with weak parity objectives can be iteratively used to solve games with bounded parity objectives. We state a key lemma that would directly lead to an algorithm (Algorithm 1) for bounded parity objectives.

Lemma 1. *For all game graphs $G = ((S, E), (S_1, S_2))$ and all priority functions p the following assertions hold.*

1. *We have $Attr_2(W_2(\text{weakcoParity}(p)), G) \subseteq W_2(\text{unbndParity}(p))$, i.e., the attractor to the winning set with objective* weakcoParity(p) *is a subset of the winning set with the unbounded parity objective* unbndParity(p)*. There is a finite-memory winning strategy π for player 2 for the objective* unbndParity(p) *from the set $Attr_2(W_2(\text{weakcoParity}(p)), G)$ such that the following conditions hold:*
   - *$|\pi| = 2$.*
   - *for all strategies σ and for all $s \in Attr_2(W_2(\text{weakcoParity}(p)), G)$ there exists $k \leq |S|$ such that for the play $\omega(s, \sigma, \pi) = \langle s_0, s_1, s_2, \ldots \rangle$ we have (a) $p(s_k)$ is odd and (b) for all $j \geq k$ if $p(s_j)$ is even, then $p(s_j) > p(s_k)$.*
2. *If $S = W_1(\text{weakParity}(p))$, then $S = W_1(\text{bndParity}(p))$ and a memoryless winning strategy exists for player 1 for the objective* bndParity(p)*.*

Proof. We prove the two cases below.

1. First observe that for a play $\omega \in \text{weakcoParity}(p)$, the smallest priority that appears in ω is odd, and let k be a position such that the smallest odd priority appear at k. Then we have $dist_k(\omega, p) = \infty$ and thus we obtain that $\omega \in \text{unbndParity}(p)$, i.e., we have $\text{weakcoParity}(p) \subseteq \text{unbndParity}(p)$. Thus we obtain that $W_2(\text{weakcoParity}(p)) \subseteq W_2(\text{unbndParity}(p))$. For a play $\omega \in \text{weakcoParity}(p)$, let k be a position such that the smallest odd

priority of ω appear at k. Given a finite prefix $w \in S^*$, for the play $w \cdot \omega$ we have $dist_{|w|+k}(w \cdot \omega, p) = \infty$, and it follows that we have

$$\{w \cdot \omega \in \Omega \mid w \in S^*, \omega \in \text{weakcoParity}(p)\} \subseteq \text{unbndParity}(p).$$

Hence we obtain that $Attr_2(W_2(\text{weakcoParity}(p)), G) \subseteq W_2(\text{unbndParity}(p))$. A witness winning strategy π for the objective unbndParity(p) for the set $Attr_2(W_2(\text{weakcoParity}(p)), G)$ is as follows: (a) play a memoryless attractor strategy to reach $W_2(\text{weakcoParity}(p))$ and (b) upon reaching $W_2(\text{weakcoParity}(p))$ switch to a memoryless winning strategy for the objective weakcoParity(p) (and such a memoryless winning strategy exists by Theorem 2). Observe that the strategy π switches between two memoryless strategies and thus we have $|\pi| = 2$. Moreover, the strategy π ensures that for all starting states $s \in Attr_2(W_2(\text{weakcoParity}(p)), G)$ and for all strategies $\sigma \in \Sigma$, a state s_k is reached within $k \leq |S|$ steps such that the priority at s_k is odd and all subsequent even priorities are greater than $p(s_k)$. Thus π is the desired winning strategy.

2. If $S = W_1(\text{weakParity}(p))$, then fix a memoryless winning strategy σ for player 1 for the objective weakParity(p) (such a strategy exists by Theorem 2). Then for all $s \in S$ and all strategies π for player 2, the following assertion hold for the play $\omega(s, \sigma, \pi) = \langle s_0, s_1, s_2, \ldots \rangle$: for $k \geq 0$, if $p(s_k)$ is odd, then there exists $k < k' \leq k + |S|$ such that $p(s_{k'}) < p(s_k)$ and $p(s_{k'})$ is even. Otherwise, there is a cycle C and a finite path w from s_k to a state in C in the graph G_σ such that for all states s' in C and w, if $p(s')$ is even, then $p(s') > p(s_k)$, where G_σ is the graph obtained from G fixing the memoryless strategy σ for player 1. Then a strategy for player 2, that executes the path w and then the cycle C for ever in G_σ, contradicts the fact that $s_k \in W_1(\text{weakParity}(p))$ and σ is a winning strategy for weakParity(p). Thus it follows that for all $s \in S$, for all strategies $\pi \in \Pi$ and for all $k \geq 0$ we have $dist_k(\omega(s, \sigma, \pi), p) \leq |S|$. Hence we have $S = W_1(\text{bndParity}(p))$.

The desired result follows. ∎

We now present an example to show that memory is need for winning strategies for objectives unbndParity(p).

Example 3 (Memory required for unbndParity(p) *objective.).* Consider the game graph shown in Fig 1 and consider the sub-game graph induced by the set $\{s_0, s_2, s_3\}$ of states. All player 1 states have only one edge and hence the hence the sub-game induced is effectively a one player game graph. We consider the objective unbndParity(p) for player 2, for the priority function p as shown in the figure. Let s_2 be the starting state and we consider two memoryless strategies for player 2:

- for the memoryless strategy $s_2 \to s_2$, the state s_2 is always visited and hence the strategy is not winning for objective unbndParity(p);
- for the memoryless strategy $s_2 \to s_3$ the sequence of priority generated is $(2 \cdot 0 \cdot 1)^\omega$, and the distance between priority 1 and priority 0 is always 2; hence the strategy is also not winning for objective unbndParity(p).

Algorithm 1 BoundedParity

```
Input: a game graph $G$ and a priority function $p$.
Output: the sets $W_1 = W_1(\text{bndParity}(p))$ and $W_2 = W_2(\text{unbndParity}(p))$.
1. $W_1 = \emptyset$; $W_2 = \emptyset$; $G^0 = G$; $i = 0$;
2. repeat
    2.1 $W_2 := W_2 \cup Attr_2(W_2^{G^i}(\text{weakcoParity}(p)), G^i)$;
    2.2 $G^{i+1} := G^i \upharpoonright (S \setminus W_2)$;
    2.3 $i := i + 1$;
    until $W_2^{G^i}(\text{weakcoParity}(p)) = \emptyset$;
3. return $(S \setminus W_2, W_2)$.
```

However, consider the following strategy π: the strategy initially chooses $s_2 \rightarrow s_3$ and after the play visits s_0, then the strategy switches and chooses $s_2 \rightarrow s_2$ forever. Hence the sequence of priority generated is $2 \cdot 0 \cdot 1 \cdot 2^\omega$, i.e., there is no priority 0 state after the priority 1 is visited. Hence the strategy is winning for the objective unbndParity(p). Thus in general winning strategies for unbndParity(p) require memory. ∎

Algorithm for bounded parity objectives. The algorithm for bounded parity objectives is obtained as follows: the algorithm takes as input a game graph G and priority function p and proceeds iteratively. We denote by G^i the game graph in iteration i. In iteration i, the algorithm computes the set $W_2^{G^i}(\text{weakcoParity}(p))$, identifies its player 2 attractor as a subset of the winning set W_2 for player 2, removes this set from the game graph, and proceeds to the next iteration. The correctness of this step follows from part 1 of Lemma 1. In every iteration at least one state is removed from the game graph and thus the algorithm proceeds for at most $|S|$ steps. Let the algorithm terminate after i-iterations, then for the sub-game graph G^i we have $W_2^{G^i}(\text{weakcoParity}(p)) = \emptyset$. Then by Theorem 2 we obtain that all states $s \in G^i$ satisfy that $s \in W_1^{G^i}(\text{weakParity}(p))$, and then by part 2 of Lemma 1 it follows that all states s in G^i satisfy that $s \in W_1^{G^i}(\text{bndParity}(p))$. Since G^i is a closed set for player 2, by Proposition 1 we obtain that all states s in G^i satisfy that $s \in W_1^G(\text{bndParity}(p))$. This proves correctness of the algorithm. The algorithm runs for at most $|S|$-iterations and by Theorem 2 each iteration can be computed in $O(|E|)$ time. This gives us the following theorem summarizing the results on games with bounded parity objectives.

Theorem 3 (Bounded parity games). *For all game graphs* $G = ((S, E), (S_1, S_2))$ *and all priority functions* p *the following assertions hold.*

1. (Determinacy). *We have* $W_1(\text{bndParity}(p)) = S \setminus W_2(\text{unbndParity}(p))$.
2. (Strategy complexity). *There exists a memoryless winning strategy* σ *for player 1 for the objective* bndParity(p) *such that for all* $s \in$

$W_1(\mathrm{bndParity}(p))$ and for all strategies π for player 2 we have $\omega(s,\sigma,\pi) \in \mathrm{bndParity}(p) \cap \mathrm{Safe}(W_1(\mathrm{bndParity}(p)))$. There exists a finite-memory winning strategy π for player 2 for the objective $\mathrm{unbndParity}(p)$, with $|\pi| = 2$. In general no memoryless winning strategy exists for player 2 for the objective $\mathrm{unbndParity}(p)$.

3. (Time complexity). *Algorithm 1 computes the sets $W_1(\mathrm{bndParity}(p))$ and $W_2(\mathrm{unbndParity}(p))$ in $O(n \cdot m)$ time, where $n = |S|$ and $m = |E|$.*

4.3 Solving games with finitary parity objectives

In this section we will show how the solution of games with bounded parity objectives can be iteratively used to solve games with finitary parity objectives. We state a key lemma that would directly lead to an algorithm (Algorithm 2) for finitary parity objectives.

Lemma 2. *For all game graphs $G = ((S,E),(S_1,S_2))$ and all priority functions p the following assertions hold.*

1. *We have $\mathit{Attr}_1(W_1(\mathrm{bndParity}(p)), G) \subseteq W_1(\mathrm{finParity}(p))$, i.e., the attractor to the winning set with objective $\mathrm{bndParity}(p)$ is a subset of the winning set with the finitary parity objective $\mathrm{finParity}(p)$. There is a memoryless winning strategy σ for player 1 from the set $\mathit{Attr}_1(W_1(\mathrm{bndParity}(p)), G)$ for the objective $\mathrm{finParity}(p)$.*
2. *If $S = W_2(\mathrm{unbndParity}(p))$, then $S = W_2(\mathrm{infParity}(p))$ and an infinite memory winning strategy π exists for player 2 for the objective $\mathrm{infParity}(p)$ such that π is finite-memory counting with $\mathrm{count}(|\pi|) = 2$.*

Proof. We prove both the cases below.

1. We have $\mathrm{bndParity}(p) \subseteq \mathrm{finParity}(p)$, and since the finitary parity objective requires the distance sequence to be ultimately bounded (i.e., bounded in the limit), it follows that the objective $\mathrm{finParity}(p)$ is independent of all finite prefixes of plays. Hence we have

$$\{w \cdot \omega \in \Omega \mid w \in S^*, \omega \in \mathrm{bndParity}(p)\} \subseteq \mathrm{finParity}(p).$$

It follows that $\mathit{Attr}_1(W_1(\mathrm{bndParity}(p)), G) \subseteq W_1(\mathrm{finParity}(p))$. A winning strategy σ is defined as follows:

   - a memoryless attractor strategy to reach $W_1(\mathrm{bndParity}(p))$; and
   - a memoryless winning strategy for objective $\mathrm{bndParity}(p)$ in $W_1(\mathrm{bndParity}(p))$ (such a memoryless strategy exists by Theorem 3).

   Also observe that the memoryless winning strategy in $W_1(\mathrm{bndParity}(p))$ ensures that the set $W_1(\mathrm{bndParity}(p))$ is never left (i.e., it ensures $\mathrm{Safe}(W_1(\mathrm{bndParity}(p))))$) and thus is independent of the memoryless attractor strategy defined for the set $\mathit{Attr}_1(W_1(\mathrm{bndParity}(p)), G) \setminus W_1(\mathrm{bndParity}(p))$. Hence σ is a memoryless winning strategy for player 1 for the objective $\mathrm{finParity}(p)$ for the set $\mathit{Attr}_1(W_1(\mathrm{bndParity}(p)), G)$.

2. If $S = W_2(\text{unbndParity}(p))$, then we produce a desired winning strategy for player 2 for the objective $\text{infParity}(p)$. Since $S = W_2(\text{unbndParity}(p))$, there exists a finite-memory winning strategy π for the objective $\text{unbndParity}(p)$ from S such that
   - $|\pi| = 2$;
   - for all strategies σ and for all $s \in S$ there exists $k \leq |S|$ such that for the play $\omega(s, \sigma, \pi) = \langle s_0, s_1, s_2, \ldots \rangle$ we have (a) $p(s_k)$ is odd and (b) for all $j \geq k$ if $p(s_j)$ is even, then $p(s_j) > p(s_k)$.

   The existence of such a strategy π follows from Lemma 1. The winning strategy π^* is obtained from π as follows:

   Step 1 Set a counter c to 1.
   Step 2 Play the strategy π for $n + c$ steps.
   Step 3 Increment c.
   Step 4 Reset the memory for π and goto to step 2.

   The strategy π^* goes through the loop (step 2—step 4) infinitely many times. For all states $s \in S$ and for all strategies σ for player 1, the strategy π at step 2 ensures that given a value c of the counter, there is a position k such that priority of $p(s_k)$ is odd and for all $k \leq k' \leq k + c$, if $p(s_{k'})$ is even, then $p(s_{k'}) > p(s_k)$, i.e., $\mathit{dist}_{k'}(\omega(s, \sigma, \pi), p) \geq c$. Let us denote by c_j the value of the counter c at the j-th iteration of the loop. The strategy π^* ensures that for all states $s \in S$, all strategies $\sigma \in \Sigma$ and all $j \geq 0$, there exists a k such that $\mathit{dist}_k(\omega(s, \sigma, \pi^*), p) \geq c_j$. Since $\lim_{j \to \infty} c_j = \infty$, it follows that for all states $s \in S$ and all strategies $\sigma \in \Sigma$ we have $\limsup_{k \to \infty} \mathit{dist}_k(\omega(s, \sigma, \pi^*), p) = \infty$, i.e., $\omega(s, \sigma, \pi^*) \in \text{infParity}(p)$.

The desired result follows. ∎

Algorithm for finitary parity objectives. The algorithm for finitary parity objectives is obtained as follows: the algorithm takes as input a game graph G and priority function p and proceeds iteratively. We denote by G^i the game graph in iteration i. In iteration i, the algorithm computes the set $W_1^{G^i}(\text{bndParity}(p))$, identifies its player 1 attractor as a subset of the winning set W_1 for player 1, removes this set from the game graph, and proceeds to the next iteration. The correctness of this step follows from part 1 of Lemma 2. In every iteration at least one state is removed from the game graph and thus the algorithm proceeds for at most $|S|$ steps. Let the algorithm terminate after i-iterations, then for the subgame graph G^i we have $W_1^{G^i}(\text{bndParity}(p)) = \emptyset$. Then by Theorem 3 we obtain that all states $s \in G^i$ satisfy that $s \in W_2^{G^i}(\text{unbndParity}(p))$, and then by part 2 of Lemma 2 it follows that all states s in G^i satisfy that $s \in W_2^{G^i}(\text{infParity}(p))$. Since G^i is a closed set for player 1, by Proposition 1 (exchanging roles of player 1 and player 2) we obtain that all states s in G^i satisfy that $s \in W_2^G(\text{infParity}(p))$. This proves correctness of the algorithm. The algorithm runs for at most $|S|$-iterations and by Theorem 3 each iteration can be computed in $O(|S| \cdot |E|)$ time. This gives us the following theorem summarizing the results on games with finitary parity objectives.

Algorithm 2 FinitaryParity

```
Input: a game graph $G$ and a priority function $p$.
Output: the sets $W_1 = W_1(\text{finParity}(p))$ and $W_2 = W_2(\text{infParity}(p))$.
1. $W_1 = \emptyset$; $W_2 = \emptyset$; $G^0 = G$; $i = 0$;
2. repeat
   2.1 $W_1 := W_1 \cup Attr_1(W_1^{G^i}(\text{bndParity}(p)), G^i)$;
   2.2 $G^{i+1} := G^i \upharpoonright (S \setminus W_1)$;
   2.3 $i := i + 1$;
   until $W_1^{G^i}(\text{bndParity}(p)) = \emptyset$;
3. return $(W_1, S \setminus W_1)$.
```

Theorem 4 (Finitary parity games). *For all game graphs $G = ((S, E), (S_1, S_2))$ and all priority functions p the following assertions hold.*

1. (Determinacy). *We have* $W_1(\text{finParity}(p)) = S \setminus W_2(\text{infParity}(p))$.
2. (Strategy complexity). *There exists a memoryless winning strategy for player 1 for the objective* finParity(p). *There exists an infinite memory winning strategy π for player 2 for the objective* infParity(p) *such that π is finite-memory counting with* count$(|\pi|) = 2$. *In general no finite-memory winning strategy exists for player 2 for the objective* infParity(p).
3. (Time complexity). *Algorithm 2 computes the sets* $W_1(\text{finParity}(p))$ *and* $W_2(\text{infParity}(p))$ *in $O(n^2 \cdot m)$ time, where $n = |S|$ and $m = |E|$.*

The existence of memoryless winning strategies for finitary parity objectives also gives the following refined characterization of the winning set, which shows that distances can be bounded by the size of the state space.

Corollary 1. *For all game graphs with n states, and all priority functions p, we have*

$$\begin{aligned} W_1(\text{finParity}(p)) &= \{s \in S \mid \exists \sigma \in \Sigma.\ \forall \pi \in \Pi.\ \limsup_{k \to \infty} dist_k(\omega(s, \sigma, \pi), p) \leq \infty\} \\ &= \{s \in S \mid \exists \sigma \in \Sigma.\ \forall \pi \in \Pi.\ \limsup_{k \to \infty} dist_k(\omega(s, \sigma, \pi), p) \leq n\}. \end{aligned}$$

5 Finitary Streett Games: Determinacy and Complexity

In this section we will study the complexity of finitary Streett games. We will first consider bounded Streett games and similar to the case of finitary parity games we will iteratively use the solution of bounded Streett games to solve finitary Streett games.

5.1 Solving games with bounded Streett objectives

The solution of bounded Streett objectives to the solution of finitary Streett objectives will play the same role as the solution of bounded parity objectives to the solution of finitary parity objectives. However, solving weak Streett games iteratively to obtain solution of bounded Streett objectives is not known (unlike the case of weak parity and bounded parity objectives). In [21] the authors studied games with *request-response specifications*, and the solution for games with request-response specifications yields a solution for games with bounded Streett objectives. The result of [21] presented a solution for request-response games based on a reduction to games with Büchi objectives. The reduction incurs a blow-up by a factor of $d \cdot 2^d$ for a set of d Streett pairs. We now summarize the result on games with bounded Streett objectives obtained from the results of [21] on request-response games.

Theorem 5 (Bounded Streett games [21]). *Given a game graph* $G = ((S, E), (S_1, S_2))$ *and a set* $P = \{(E_1, F_1), \dots, (E_d, F_d)\}$ *of* d *Streett pairs, the following assertions hold.*

1. (Determinacy). *We have* $W_1(\text{bndStreett}(P)) = S \setminus W_2(\text{unbndStreett}(P))$.
2. (Strategy complexity). *There exists a finite-memory winning strategy* σ *for player 1 for the objective* $\text{bndStreett}(P)$ *such that the following conditions hold:*
   (a) $|\sigma| = d \cdot 2^d$*;*
   (b) for all $s \in W_1(\text{bndStreett}(P))$*, for all strategies* π *and for all* $k \geq 0$*, we have* $dist_k(\omega(s, \sigma, \pi), P) \leq |S| \cdot d \cdot 2^d$.

   In general winning strategies for player 1 for the objective $\text{bndStreett}(P)$ *requires* $\lfloor \frac{d}{3} \rfloor \cdot 2^{\lfloor \frac{d}{3} \rfloor}$ *memory. There exists a finite-memory winning strategy* π *for player 2 for the objective* $\text{unbndStreett}(P)$ *such that* $|\pi| = d \cdot 2^d$.
3. (Time complexity). *The sets* $W_1(\text{bndStreett}(P))$ *and* $W_2(\text{unbndStreett}(P))$ *can be computed in* $O(n \cdot m \cdot 4^d \cdot d^2)$*, where* $n = |S|$ *and* $m = |E|$.

5.2 Solving games with finitary Streett objectives

We now show that the solution of games with bounded Streett objectives can be used iteratively to solve games with finitary Streett objectives. We now prove the following lemma that would directly lead to an algorithm for finitary Streett objectives. The role of Lemma 3 to obtain Algorithm 3 for finitary Streett games is same as the role of Lemma 2 to obtain Algorithm 2 for finitary parity games.

Lemma 3. *Given a game graphs* $G = ((S, E), (S_1, S_2))$ *and a set* $P = \{(E_1, F_1), \dots, (E_d, F_d)\}$ *of* d *Streett pairs, the following assertions hold.*

1. *We have* $Attr_1(W_1(\text{bndStreett}(P)), G) \subseteq W_1(\text{finStreett}(P))$*, i.e., the attractor to the winning set with objective* $\text{bndStreett}(P)$ *is a subset of the winning set with the finitary Streett objective* $\text{finStreett}(P)$*. There is a finite-memory winning strategy* σ *for player 1 from the set* $Attr_1(W_1(\text{bndStreett}(P), G)$ *for the objective* $\text{finStreett}(P)$ *such that* $|\sigma| = d \cdot 2^d$.
2. *If* $S = W_2(\text{unbndStreett}(P))$*, then* $S = W_2(\text{infStreett}(P))$ *and an infinite-memory winning strategy* π *exists for player 2 for the objective* $\text{infStreett}(P)$ *such that* π *is finite-memory counting with* $\text{count}(|\pi|) = d \cdot 2^d$.

Proof. We prove both the cases below.

1. We have bndStreett$(P) \subseteq$ finStreett(P), and since the finitary Streett objective requires the distance sequence to be ultimately bounded (i.e., bounded in the limit), it follows that the objective finStreett(P) is independent of all finite prefixes of plays. Hence we have
$$\{w \cdot \omega \in \Omega \mid w \in S^*, \omega \in \text{bndStreett}(P)\} \subseteq \text{finStreett}(P).$$
It follows that $Attr_1(W_1(\text{bndStreett}(P)), G) \subseteq W_1(\text{finStreett}(P))$. A winning strategy σ is defined as follows:
   - a memoryless attractor strategy to reach $W_1(\text{bndStreett}(P))$; and
   - a finite-memory winning strategy for objective bndStreett(P) in $W_1(\text{bndStreett}(P))$ such that $|\sigma| = d \cdot 2^d$ (such a strategy exists by Theorem 5).

   The winning strategy in $W_1(\text{bndStreett}(P))$ ensures that the set $W_1(\text{bndStreett}(P))$ is never left (i.e., it ensures Safe$(W_1(\text{bndParity}(p)))$) and thus is independent of the memoryless attractor strategy defined for the set $Attr_1(W_1(\text{bndStreett}(P)), G) \setminus W_1(\text{bndStreett}(P))$. Hence σ is a finite-memory winning strategy for player 1 for the objective finStreett(P) for the set $Attr_1(W_1(\text{bndStreett}(P)), G)$, with $|\sigma| = d \cdot 2^d$.
2. If $S = W_2(\text{unbndStreett}(P))$, then we produce a desired winning strategy for player 2 for the objective infStreett(P). Since $S = W_2(\text{unbndParity}(p))$, there exists a finite-memory winning strategy π for the objective unbndParity(P) from S. The existence of such a strategy π follows from Theorem 5. The winning strategy π^* is obtained from π as follows:
Step 1 Set a counter c to 1.
Step 2 Play the strategy π until there is a sequence such that there is a state $s_k \in F_j$ and for all $k < k' \leq k + c$ we have $s_{k'} \notin E_j$, for some $1 \leq j \leq d$.
Step 3 Increment c.
Step 4 Reset the memory of π and goto to step 2.
Given a strategy σ for player 1 and a state $s \in S$ we first argue that the strategy π^* goes through the loop (step 2—step 4) infinitely often. Consider the play $\omega(s, \sigma, \pi^*) = \langle s_0, s_1, s_2, \ldots \rangle$. Assume towards contradiction that the play gets stuck in step 2 in iteration i, then let ℓ be the length of the play before iteration i. Then the strategy σ' that plays like σ but appending the prefix $\langle s_0, s_1, \ldots, s_{\ell-1} \rangle$ ensures that in the play $\omega(s_\ell, \sigma', \pi)$ for all F_j states, there is a E_j state with in $i + 1$ steps, i.e., $\omega(s_\ell, \sigma', \pi) \in$ bndStreett(P), this contradicts that π is a winning strategy for unbndStreett(P). Hence, the strategy π^* goes through the loop (step 2—step 4) infinitely many times, and then similar to the proof of Lemma 2 we obtain that $\limsup_{k \to \infty} dist_k(\omega(s, \sigma, \pi^*), P) = \infty$, i.e., $\omega(s, \sigma, \pi^*) \in$ unbndStreett(P). Moreover, given a value c for the counter, the strategy π (obtained from Theorem 5) can be played for $|S| \cdot d \cdot 2^d \cdot (c + 1)$ steps and the condition for step 2 can be satisfied. Hence a winning strategy π^* exists for objective unbndStreett(P) such that count$(|\pi^*|) = d \cdot 2^d$.

The desired result follows. ∎

Example 4 (Lower bound on memory). We now present an example to show that for finitary Streett objectives with $2d$ Streett pairs, winning strategies in general require at least 2^d memory. We consider a game graph $G = ((S, E), (S_1, S_2))$ with a set P of $2d$ Streett pairs as follows.

1. *State space.* $S = \{s_d, \widehat{s}_d\} \cup \{s_i, s_i^+, s_i^-, \widehat{s}_i, \widehat{s}_i^+, \widehat{s}_i^- \mid 0 \leq i \leq d-1\}$.
2. *State space partition.* The state space partition into (S_1, S_2) is as follows:
$$S_1 = \{\widehat{s}_i, \widehat{s}_i^+, \widehat{s}_i^- \mid 0 \leq i \leq d-1\} \cup \{\widehat{s}_d\};$$
$$S_2 = \{s_i, s_i^+, s_i^- \mid 0 \leq i \leq d-1\} \cup \{s_d\}.$$
3. *Edges.* The set of edges are as follows:
$$\begin{aligned} E = &\{(s_0, s_0), (s_d, \widehat{s}_0), (\widehat{s}_d, s_0)\} \\ &\cup \{(s_i, s_i^+), (s_i, s_i^-) \mid 0 \leq i \leq d-1\} \cup \{(s_i^+, s_{i+1}), (s_i^-, s_{i+1}) \mid 0 \leq i \leq d-1\} \\ &\cup \{(\widehat{s}_i, \widehat{s}_i^+), (\widehat{s}_i, \widehat{s}_i^-) \mid 0 \leq i \leq d-1\} \cup \{(\widehat{s}_i^+, \widehat{s}_{i+1}), (\widehat{s}_i^-, \widehat{s}_{i+1}) \mid 0 \leq i \leq d-1\}. \end{aligned}$$
4. *Streett pairs.* The set P, that consists of $2d$ Streett pairs, is as follows:
$$P = \{(E_0^+, F_0^+), (E_1^+, F_1^+), \ldots, (E_{d-1}^+, F_{d-1}^+), (E_0^-, F_0^-), (E_1^-, F_1^-), \ldots, (E_{d-1}^-, F_{d-1}^-)\}.$$
For $0 \leq i \leq d-1$ we have $F_i^+ = \{s_i^+\}, F_i^- = \{s_i^-\}$; and $E_i^+ = \{\widehat{s}_i^+\}, E_i^- = \{\widehat{s}_i^-\}$.

The intuitive description of the game is as follows. For $0 \leq i \leq d-1$, at state s_i player 2 can choose to go to s_i^+ or s_i^- and thus generate a state in either F_i^+ or F_i^-; at state s_0 there is a self-loop to ensure that player 2 also have the choice to stay in s_0. For $0 \leq i \leq d-1$ from states s_i^+ and s_i^- the next state is s_{i+1}, and the next state of s_d is $\widehat{s}_0$. For $0 \leq i \leq d-1$, at state $\widehat{s}_i$ player 1 can choose to go to $\widehat{s}_i^+$ or $\widehat{s}_i^-$ and thus generate a state in either E_i^+ or E_i^-. For $0 \leq i \leq d-1$ from states $\widehat{s}_i^+$ and $\widehat{s}_i^-$ the next state is $\widehat{s}_{i+1}$, and the next state of $\widehat{s}_d$ is s_0. A pictorial description of the game is shown in Fig 3.

Winning strategy. We consider the objective finStreett(P) and starting state s_0, and show that a winning strategy σ with $|\sigma| = 2^d$ exists. The strategy σ at state $\widehat{s}_i$ chooses $\widehat{s}_i^\alpha$, if the last choice at state s_i is s_i^α, for $\alpha \in \{+, -\}$. In other words, the strategy matches each choice of F_j^+ or F_j^- of player 2 by a matching choice of E_j^+ or E_j^-. Thus player 1 can ensure that the distance sequence is bounded by $2d + 2$, and σ is a winning strategy. We now argue that winning strategies require at least 2^d memory. A spoiling strategy π for player 2 against strategies with memory less than 2^d is as follows. The strategy π is played in rounds. With memory less than 2^d player 1 cannot remember all sequences $f_0, f_1, \ldots, f_{d-1}$, where for $0 \leq i \leq d-1$ we have $f_i \in \{F_i^+, F_i^-\}$. The strategy for player 2 in round j waits in state s_0 (by the choice of self-loop) for j-steps, then start generating all 2^d possible choices of sequences $f_0, f_1, \ldots, f_{d-1}$, where

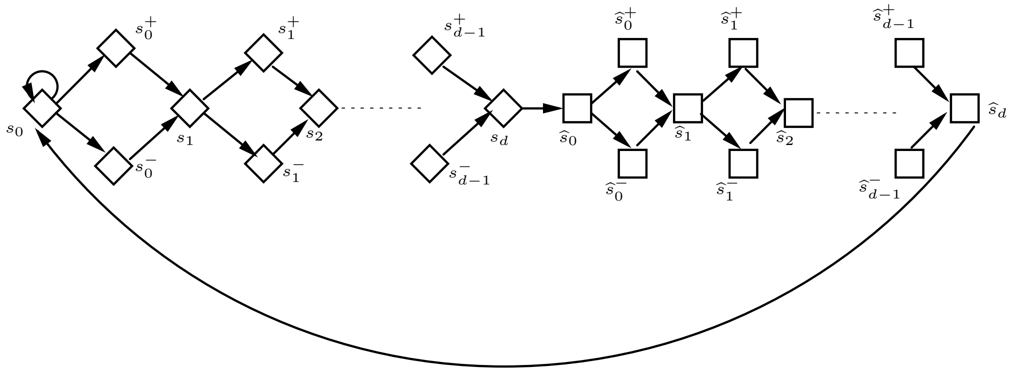

Fig. 3. Game graph for lower bound on memory.

Algorithm 3 FinitaryStreett

```
Input: a game graph $G$ and a set $P = \{(E_1, F_1), (E_2, F_2), \ldots, (E_d, F_d)\}$ of $d$ Streett pairs.
Output: the sets $W_1 = W_1(\text{finStreett}(P))$ and $W_2 = W_2(\text{infStreett}(P))$.

1. $W_1 = \emptyset$; $W_2 = \emptyset$; $G^0 = G$; $i = 0$;
2. repeat
   2.1 $W_1 := W_1 \cup Attr_1(W_1^{G^i}(\text{bndStreett}(P)), G^i)$;
   2.2 $G^{i+1} := G^i \upharpoonright (S \setminus W_1)$;
   2.3 $i := i + 1$;
   until $W_1^{G^i}(\text{bndStreett}(p)) = \emptyset$;
3. return $(W_1, S \setminus W_1)$.
```

for $0 \leq i \leq d-1$ we have $f_i \in \{F_i^+, F_i^-\}$. Whenever player 1 fails to match the sequence by a corresponding sequence of E_i^+ and E_i^-, then player 2 moves to round $j+1$. With memory less than 2^d player 1 cannot match every sequence, and hence limsup of the distance sequence is ∞, i.e., the finitary Streett objective is violated. ∎

Algorithm for finitary Streett games. As we derived from Lemma 2 the algorithm (Algorithm 2) for finitary parity games, from Lemma 3 we obtain Algorithm 3 for finitary Streett games. The correctness follows from Lemma 3 and the arguments similar to correctness of Algorithm 2. We now summarize the results on finitary Streett games in the following theorem.

Theorem 6 (Finitary Streett games). *Given a game graph $G = ((S, E), (S_1, S_2))$ and a set $P = \{(E_1, F_1), \ldots, (E_d, F_d)\}$ of d Streett pairs, the following assertions hold.*

1. (Determinacy). *We have $W_1(\text{finStreett}(P)) = S \setminus W_2(\text{infStreett}(P))$.*

2. (Strategy complexity). *There exists a finite-memory winning strategy σ for player 1 for the objective* $\mathrm{finStreett}(P)$ *such that $|\sigma| = d \cdot 2^d$. In general no memoryless winning strategy exists for player 1 for the objective* $\mathrm{finStreett}(P)$*. In general winning strategies for player 1 for the objective* $\mathrm{finStreett}(P)$ *require $2^{\lfloor \frac{d}{2} \rfloor}$ memory. There exists an infinite-memory winning strategy π for player 2 for the objective* $\mathrm{infStreett}(P)$ *such that π is finite-memory counting with* $\mathrm{count}(|\pi|) = d \cdot 2^d$*. In general no finite-memory winning strategy exists for player 2 for the objective* $\mathrm{infStreett}(P)$*.*

3. (Time complexity). *Algorithm 3 computes the sets* $W_1(\mathrm{finStreett}(P))$ *and* $W_2(\mathrm{infStreett}(P))$ *in $O(n^2 \cdot m \cdot d^2 \cdot 4^d)$ time, where $n = |S|$ and $m = |E|$.*

The winning strategy for finitary Streett objectives with d Streett pairs requires $d \cdot 2^d$ memory, this is in contrast to classical Streett objectives that require $d!$ memory [7, 10], and weak Streett objectives that require 2^d memory [17]. The winning strategy for finitary Streett objectives is obtained by composing winning strategies for bounded Streett objectives, that ensured that for every occurrence of a state in F_j, a state in E_j appears with in $n \cdot d \cdot 2^d$ steps, where n is the number of states and d is the number of Streett pairs. This gives us the following refined characterization of the winning set.

Corollary 2. *Given a game graph $G = ((S, E), (S_1, S_2))$ and a set $P = \{(E_1, F_1), \ldots, (E_d, F_d)\}$ of d Streett pairs,*

$$\begin{aligned} W_1(\mathrm{finStreett}(P)) &= \{s \in S \mid \exists \sigma \in \Sigma.\ \forall \pi \in \Pi.\ \limsup_{k \to \infty} \mathit{dist}_k(\omega(s, \sigma, \pi), P) \leq \infty\} \\ &= \{s \in S \mid \exists \sigma \in \Sigma.\ \forall \pi \in \Pi.\ \limsup_{k \to \infty} \mathit{dist}_k(\omega(s, \sigma, \pi), P) \leq n \cdot d \cdot 2^d\}; \end{aligned}$$

where $n = |S|$.

It follows from Theorem 6 that whether a state lies in $W_1(\mathrm{finStreett}(P))$ can be decided in EXPTIME. We now prove a lower bound for the problem.

Lower bound for finitary Streett games. We show that given a game graph G and finitary Streett objective $\mathrm{finStreett}(P)$ the problem of deciding whether $s \in W_1(\mathrm{finStreett}(P))$ is NP-hard. We present a reduction of the 3-SAT problem. Let $\varphi = C_1 \wedge C_2 \ldots \wedge C_m$ be a 3-SAT formula with clauses $C_1, C_2, \ldots, C_m$ over variables $x_1, x_2, \ldots, x_n$. For a clause C_i and a literal x_j we write $x_j \in C_i$ if x_j appears in C_i, and similarly, we write $\neg x_j \in C_i$ if $\neg x_j$ appears in C_i. We construct a game graph $G = ((S, E), (S_1, S_2))$ with a finitary Streett objective as follows.

1. *State space.* The state space S is defined as follows:

$$\begin{aligned} S = {} & \{c_0, c_1, c_2, \ldots, c_m, c_{m+1}\} \\ & \cup \{x_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\} \ \cup \ \{\neg x_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\} \\ & \cup \{\widehat{x}_i, \widehat{x}_i^+, \widehat{x}_i^- \mid 1 \leq i \leq n\} \cup \{\widehat{x}_{n+1}\}. \end{aligned}$$

2. *State space partition.* $S_2 = \{c_0\}$ and $S_1 = S \setminus S_2$.

3. *Edges.* The set E of edges is as follows:

$$\begin{aligned} E = & \{(c_0, c_0), (c_0, c_1), (c_{m+1}, \widehat{x}_1), (\widehat{x}_{n+1}, c_0)\} \\ & \cup \{(c_i, x_{i,j}) \mid x_j \in C_i, 1 \leq i \leq m\} \cup \{(c_i, \neg x_{i,j}) \mid \neg x_j \in C_i, 1 \leq i \leq m\} \\ & \cup \{(x_{i,j}, c_{i+1}) \mid 1 \leq i \leq m, 1 \leq j \leq n\} \cup \{(\neg x_{i,j}, c_{i+1}) \mid 1 \leq i \leq m, 1 \leq j \leq n\} \\ & \cup \{(\widehat{x}_i, \widehat{x}_i^+) \mid 1 \leq i \leq n\} \cup \{(\widehat{x}_i, \widehat{x}_i^-) \mid 1 \leq i \leq n\} \\ & \cup \{(\widehat{x}_i^+, \widehat{x}_{i+1}) \mid 1 \leq i \leq n\} \cup \{(\widehat{x}_i^-, \widehat{x}_{i+1}) \mid 1 \leq i \leq n\} \end{aligned}$$

   The intuitive interpretation of the edges are as follows. At state c_0 player 2 can either stay at c_0 or else proceed to c_1. For $1 \leq i \leq m$, the state c_i correspond to the clause C_i and the successor of c_i consists of states $x_{i,j}$ (resp. $\neg x_{i,j}$) such that x_j (resp. $\neg x_j$) appear in C_i (i.e., the choice of literals that makes C_i true). For $1 \leq i \leq m$, the successor state of states $x_{i,j}$ and $\neg x_{i,j}$ is the state c_{i+1}. From state c_{m+1} the next state is $\widehat{x}_1$. For $1 \leq i \leq n$, at state $\widehat{x}_i$ there is a choice between $\widehat{x}_i^+$ (that will correspond to the choice of literal x_i) and $\widehat{x}_i^-$ (that will correspond to the choice of literal $\neg x_i$). The next state for states $\widehat{x}_i^+$ and $\widehat{x}_i^-$ is $\widehat{x}_{i+1}$. From the state $\widehat{x}_{n+1}$ the next state is c_0.

4. *Streett pairs.* The Streett pairs

$$P = \{(E_1^+, F_1^+), (E_2^+, F_2^+), \ldots, (E_n^+, F_n^+), (E_1^-, F_1^-), (E_2^-, F_2^-), \ldots, (E_n^-, F_n^-)\}$$

   is described as follows:

$$F_i^+ = \{x_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}; \qquad F_i^- = \{\neg x_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$$

$$E_i^+ = \{\widehat{x}_i^+\}; \qquad E_i^- = \{\widehat{x}_i^-\}$$

We will consider the game graph with c_0 as the starting state and the objective $\text{finStreett}(P)$.

Satisfiability implies winning. If φ is satisfiable, then consider a satisfiable assignment A, i.e., A assigns truth value *true* or *false* to every variable and satisfies every clause of φ. For $1 \leq i \leq n$, let us denote $\text{Choice}(i)$ as follows:

$$\text{Choice}(i) = \begin{cases} x_i & \text{if } A(x_i) = \textit{true}; \\ \neg x_i & \text{if } A(x_i) = \textit{false}. \end{cases}$$

A winning strategy σ for player 1 for $\text{finStreett}(P)$ is as follows:

- For $1 \leq i \leq m$, there exists $1 \leq j \leq n$, such that $\text{Choice}(j) \in C_i$ (since A satisfies clause C_i). For $1 \leq i \leq m$, and a state c_i, pick a j such that $\text{Choice}(j) \in C_i$, the strategy σ chooses the successor $x_{i,j}$ if $\text{Choice}(j) = x_j$, else the successor $\neg x_{i,j}$ is chosen.
- For $1 \leq i \leq n$, for a state $\widehat{x}_i$ the strategy σ chooses the successor $\widehat{x}_i^+$ if $\text{Choice}(i) = x_i$ and $\widehat{x}_i^-$ otherwise.

Since A is a satisfying assignment and a consistent assignment, it follows that given the strategy σ, states in both F_i^+ and F_i^- cannot be visited, for all $1 \leq i \leq n$. Moreover, if F_i^+ is visited then the choice at $\widehat{x}_i$ is $\widehat{x}_i^+$, and if F_i^- is visited then the choice at $\widehat{x}_i$ is $\widehat{x}_i^-$. It follows that (a) if a state in F_i^+ is visited, then a state in E_i^+ is visited within $2n + 2m + 2$ steps; and (b) if a state in F_i^- is visited, then a state in E_i^- is visited within $2n + 2m + 2$ steps. It follows that $c_0 \in W_1(\text{finStreett}(P))$, i.e., if φ is satisfiable, then $s \in W_1(\text{finStreett}(P))$.

Not satisfiable implies not winning. Suppose φ is not satisfiable, then for any strategy σ for player 1 there must exist c_i and c_k such that the choice at c_i is $x_{i,j}$ and the choice at c_k is $\neg x_{i,j}$ (this is because if φ is not satisfiable, to satisfy all clauses inconsistent assignments must be chosen). That is both states in F_j^+ and F_j^- are visited. If the choice at $\widehat{x}_i$ is $\widehat{x}_i^+$, then a state in E_i^- is not visited, and if the choice is $\widehat{x}_i^-$, then a state in E_i^+ is not visited. A winning strategy for player 2 for infStreett(P) is as follows: the strategy is played in rounds, and in round i, player 2 stays in c_0 for i steps, then move to c_1 and proceed to round $i+1$. This shows that $c_0 \not\in W_1(\text{finStreett}(P))$, i.e., if φ is not satisfiable, then $c_0 \not\in W_1(\text{finStreett}(P))$. This completes the reduction. A similar reduction works for bounded Streett objectives on game graphs with $S_2 = \emptyset$ (i.e., graphs where only player 1 makes choices). The reduction is as above with the following modification: remove the self-loop from c_0, convert c_0 to a player 1 state, remove the edge from $\widehat{x}_{n+1}$ to c_0, and instead add a self-loop at $\widehat{x}_{n+1}$. Then for the modified game graph with $S_2 = \emptyset$, we have φ is satisfiable iff the state c_0 is winning for the bounded Streett objective bndStreett(P). This gives us the following theorem.

Theorem 7 (Computational complexity). *The following assertions hold.*

1. *Given a game graph $G = ((S, E), (S_1, S_2))$, a state $s \in S$ and a finitary Streett objective* finStreett(P) *the decision problem of whether $s \in W_1(\text{finStreett}(P))$ is NP-hard and can be decided in EXPTIME.*
2. *Given a game graph $G = ((S, E), (S_1, S_2))$, a state $s \in S$ and a bounded Streett objective* bndStreett(P) *the decision problem of whether $s \in W_1(\text{bndStreett}(P))$ is NP-hard and can be decided in EXPTIME. The decision problem is NP-hard even for the special case of game graphs with $S_2 = \emptyset$.*

6 Conclusion

We studied games with finitary parity and Streett objectives: we proved determinacy, presented algorithms to solve these games, and characterized the memory requirements of winning strategies for both players. The algorithm for finitary parity games has a polynomial time complexity. For finitary Streett games, we give an EXPTIME algorithm and show that the problem of deciding whether a state is winning for player 1 is NP-hard. The exact complexity of finitary Streett games remains open. A polynomial-time reduction of finitary parity objectives to weak Streett objectives remains also open; such a reduction would imply that finitary Streett games can be solved in PSPACE. The algorithm we presented for finitary Streett games is a polynomial-time reduction of finitary Streett objectives to bounded Streett objectives; thus an NP (resp. PSPACE) upper bound for bounded Streett objectives would imply an NP (resp. PSPACE) upper bound for finitary Streett objectives.

References

1. B. Alpern and F.B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.
2. R. Alur and T.A. Henzinger. Finitary fairness. In *Proc. Logic in Computer Science*, pages 52–61. IEEE Computer Society, 1994.
3. R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49:672–713, 2002.
4. K. Chatterjee. Linear Time Algorithm for Weak Parity Games Technical Report No. UCB/EECS-2006-153. University of California at Berkeley, 2006.
5. K. Chatterjee and T.A. Henzinger. Finitary winning in ω-regular games. In *Proc. of Tools and Algorithms for the Construction and Analysis of Systems*, pages 257–271, LNCS 3920, Springer, 2006.
6. N. Dershowitz, D.N. Jayasimha, and S. Park. Bounded fairness. In *Verification: Theory and Practice*, pages 304–317. LNCS 2772, Springer, 2003.
7. S. Dziembowski, M. Jurdziński and I. Walukiewicz. How much memory is needed to win infinite games ? In *Proc. Logic In Computer Science*, pages 99–110, IEEE Computer Society, 1997.
8. E.A. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proc. Foundations of Computer Science*, pages 328–337. IEEE Computer Society, 1988.
9. Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proc. Symp. Theory of Computing*, pages 60–65. ACM, 1982.
10. F. Horn. Streett games on finite graphs. *Workshop on Games in Design and Verification*, 2005.
11. F. Horn. Faster algorithms for finitary games. In *Proc. of Tools and Algorithms for the Construction and Analysis of Systems*, LNCS 4424, Springer, 2007.
12. M. Jurdzinski. Small progress measures for solving parity games. In *Symp. Theoretical Aspects of Computer Science*, pages 290–301. LNCS 1770, Springer, 2000.
13. M. Jurdziński, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, pages 117–123, ACM/SIAM, 2006.
14. C. Löding and W. Thomas. Alternating automata and logics over infinite words. In *Proc. IFIP International Conference on Theoretical Computer Science*, pages 521–535, LNCS 1872, Springer, 2000.
15. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification.* Springer, 1992.
16. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.
17. J. Neumann, A. Szepietowski and I. Walukiewicz. Complexity of weak acceptance conditions in tree automata. In *Information Processing Letters*, volume 84, p181–187, Elsevier, 2002.

18. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. Principles of Programming Languages*, pages 179–190. ACM, 1989.
19. P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM J. Control and Optimization*, 25:206–230, 1987.
20. W. Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.
21. N. Wallmeier, P. Hutten and W. Thomas. Symbolic synthesis of finite-state controllers for request-response specifications. In *Proc. Conference on Implementation and Application of Automata*, pages 11–22, LNCS 2759, Springer, 2003